

Agenda Item:

**Source:** SAMSUNG Electronics Co. and LGIC

**Title:** Text proposal for turbo codes and rate matching in TS 25.212, TS 25.222 (rev. of R1-99b56)

**Document for:** Decision

## 1. Introduction

Rate matching algorithm for Turbo codes proposed by Samsung and LGIC has been approved as a working assumption in Ad Hoc 5 meeting. This document proposes a revised text proposal for the rate matching and its corresponding mother code rate of turbo codes for uplink and downlink[1-3].

## 2. Text proposal for TS 25.212

----- Start text proposal -----

### 4.2.3. Channel coding

The following channel coding schemes can be applied to transport channels:

- Convolutional coding
- Turbo coding
- No channel coding

**Table 1: Error Correction Coding Parameters**

Transport channel type	Coding scheme	Coding rate
BCH	Convolutional code	1/2
PCH		
FACH		
RACH		
DCH		
DCH	Turbo code	1/3, <del>1/2</del> or no coding

NOTE 1: The exact physical layer encoding/decoding capabilities for different code types are FFS.

NOTE 2: In the UE the channel coding capability should be linked to the terminal class.

----- End text proposal -----

----- Start text proposal -----

#### 4.2.3.2.1. Turbo coder

NOTE: 4-state SCCC is not included in Release-99. It needs to be clarified from TSG-SA what are the service specifications with respect to different quality of services. The performance below BER of  $10^{-6}$  need to be studied if there is a requirement for this quality of services of physical layer.

For data services requiring quality of service between  $10^{-3}$  and  $10^{-6}$  BER inclusive, parallel concatenated convolutional code (PCCC) with 8-state constituent encoders is used.

The 8-state PCCC and the 4-state SCCC -are described below.

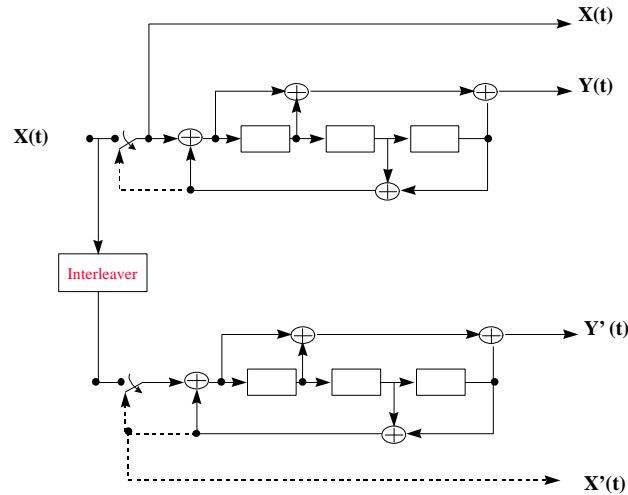
The transfer function of the 8-state constituent code for PCCC is

$$G(D) = \begin{bmatrix} 1, & n(D) \\ 1, & d(D) \end{bmatrix}$$

where,

$$d(D) = 1 + D^2 + D^3$$

$$n(D) = 1 + D + D^3$$



**Figure 1: Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)**

The initial value of the shift registers of the PCCC encoder shall be all zeros.

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate 1/3 ~~or 1/2~~. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc. ~~For rate 1/2, the parity bits produced by the constituent encoders are alternately punctured to produce the output sequence X(0), Y(0), X(1), Y'(1), X(2), Y(2), X(3), Y'(3), etc.~~

The SCCC is a rate 1/3 SCCC, The outer code of the SCCC is a rate 2/3 obtained by puncturing a rate 1/2 code with generating matrix

$$G^{(o)}(Z) = (1, (1 + Z^2) / (1 + Z + Z^2))$$

----- End text proposal -----

----- Start text proposal -----

#### 4.2.6. Rate matching

Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

The number of bits on a transport channel can vary between different transmission time intervals. In the downlink the transmission is interrupted if the number of bits is lower than maximum. When the number of bits between different transmission time intervals in uplink is changed, bits are repeated or punctured to ensure that the total bit rate after second multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

Notation used in Section 4.2.6 and subsections:

$N_{ij}$ : Number of bits in a radio frame before rate matching on transport channel  $i$  with transport format combination  $j$ .

$N_{ij}^{TTI}$ : Number of bits in a transmission time interval before rate matching on transport channel  $i$  with transport format  $j$ .

$\Delta N_{ij}$ : If positive - number of bits that should be repeated in each radio frame on transport channel  $i$  with transport format

combination  $j$ .

If negative - number of bits that should be punctured in each radio frame on transport channel  $i$  with transport format combination  $j$ .

$\Delta N_{ij}^{TTI}$ : If positive - number of bits to be repeated in each transmission time interval on transport channel  $i$  with transport format  $j$ .

If negative - number of bits to be punctured in each transmission time interval on transport channel  $i$  with transport format  $j$ .

$RM_i$ : Semi-static rate matching attribute for transport channel  $i$ . Signalled from higher layers.

$PL$ : Puncturing limit for uplink. This value limits the amount of puncturing that can be applied in order to avoid multicode or to enable the use of a higher spreading factor. Signalled from higher layers.

$N_{data,j}$ : Total number of bits that are available for the CCTrCH in a radio frame with transport format combination  $j$ .

$T$ : Number of transport channels in the CCTrCH.

$Z_{mj}$ : Intermediate calculation variable.

$F_i$ : Number of radio frames in the transmission time interval of transport channel  $i$ .

$k$ : Radio frame number in the transmission time interval of transport channel  $i$  ( $0 \leq k < F_i$ ).

$q$ : Average puncturing distance.

$I_F(k)$ : The inverse interleaving function of the 1<sup>st</sup> interleaver (note that the inverse interleaving function is identical to the interleaving function itself for the 1<sup>st</sup> interleaver).

$S(k)$ : The shift of the puncturing pattern for radio frame  $k$ .

$TF_i(j)$ : Transport format of transport channel  $i$  for the transport format combination  $j$ .

$N_{cs,j}$ : Number of bits before rate matching for each rate matching algorithm block.

$N_{is,j}$ : Number of bits after rate matching for each rate matching algorithm block.

The following relations are used when calculating the rate matching pattern:

$$Z_{0,j} = 0$$

$$Z_{mj} = \left\lfloor \frac{\sum_{i=1}^m RM_i \cdot N_{ij}}{T} \cdot N_{data,j} \right\rfloor \quad \text{for all } m = 1 .. T, \text{ where } \lfloor \cdot \rfloor \text{ means round downwards}$$

$$\Delta N_{ij} = Z_{ij} - Z_{i-1,j} - N_{ij} \quad \text{for all } i = 1 .. T$$

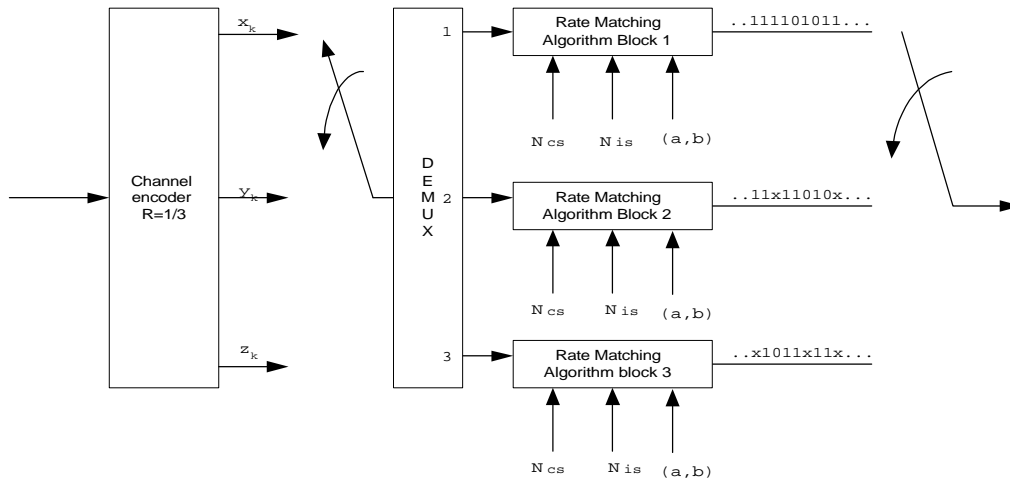


Figure 4-1. Overall rate matching block diagram before first interleaving where x denotes punctured bit.

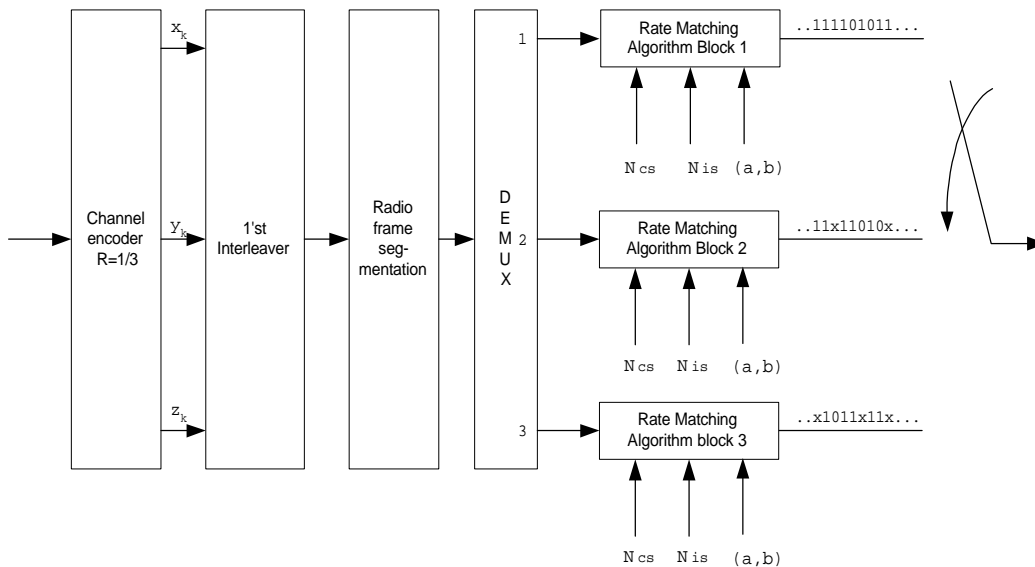


Figure 4-2. Overall rate matching block diagram after first interleaving where x denotes punctured bit.

#### 4.2.6.1. Determination of rate matching parameters in uplink

In uplink puncturing can be used to avoid multicode or to enable the use of a higher spreading factor. The maximum amount of puncturing that can be applied is signalled at connection setup from higher layers and denoted by  $PL$ . The number of available bits in the radio frames for all possible spreading factors is given in [2]. Denote these values by  $N_{256}$ ,  $N_{128}$ ,  $N_{64}$ ,  $N_{32}$ ,  $N_{16}$ ,  $N_8$ , and  $N_4$ , where the index refers to the spreading factor. The possible values of  $N_{data}$  then are  $\{N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, N_4, 2N_4, 3N_4, 4N_4, 5N_4, 6N_4\}$ . Depending on the UE capabilities, the supported set of  $N_{data}$ , denoted SET0, can be a subset of  $\{N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, N_4, 2N_4, 3N_4, 4N_4, 5N_4, 6N_4\}$ .  $N_{data,j}$  for the transport format combination j is determined by executing the following algorithm:

$$SET1 = \{ N_{data} \text{ in SET0 such that } N_{data} - \sum_{i=1}^T \frac{RM_i}{\min\{RM_i\}} \cdot N_{ij} \text{ is non negative} \}$$

If the smallest element of SET1 requires just one DPDCH then

$N_{data,j} = \min \text{SET1}$   
else

SET2 = {  $N_{data}$  in SET0 such that  $N_{data} - PL \cdot \sum_{i=1}^T \frac{RM_i}{\min\{RM_l\}} \cdot N_{ij}$  is non negative }

Sort SET2 in ascending order

$N_{data} = \min \text{SET2}$

While  $N_{data}$  is not the max of SET2 and the follower of  $N_{data}$  requires no additional DPDCH do

$N_{data} = \text{follower of } N_{data} \text{ in SET2}$

End while

$N_{data,j} = N_{data}$

End if

The number of bits to be repeated or punctured,  $DN_{ij}$ , within one radio frame for each transport channel  $i$  is calculated with the relations given in Section 4.2.6 for all possible transport format combinations  $j$  and selected every radio frame. For each radio frame, the rate-matching pattern is calculated with the algorithm in Section 4.2.6.3, where  $DN = DN_{ij}$  and  $N = N_{ij}$ .  
Additionally, the following parameters are needed:

#### 4.2.6.1.1. Parameters for convolutional codes

'X' coded bits sequence from output 0 of convolutional coder in section 4.2.3.1.1.

'Y' coded bits sequence from output 1 of convolutional coder in section 4.2.3.1.1.

'Z' coded bits sequence from output 2 of convolutional coder in section 4.2.3.1.1.

$$N_{cs} = N_{ij}$$

$$N_{is} = N_{ij} + DN_{ij}$$

$(a,b) = (2,1)$  for the rate matching algorithm block 1.

$q = \mathring{e}N_{ij} / (\mathring{O}DN_{ij}\mathring{O})\mathring{u}$ , where  $\mathring{e} \mathring{u}$  means round downwards and  $\mathring{O}\mathring{O}$  means absolute value.

if  $q$  is even  
then  $q' = q - \text{gcd}(q, F_i)/F_i$  -- where  $\text{gcd}(q, F_i)$  means greatest common divisor of  $q$  and  $F_i$   
-- note that  $q'$  is not an integer, but a multiple of  $1/8$

else

$$q' = q$$

endif

for  $l = 0$  to  $F_i - 1$   
 $S(I_F(\mathring{e}' * q' \mathring{u} \bmod F_i)) = (\mathring{e}' * q' \mathring{u} \text{ div } F_i)$  -- where  $\mathring{e}' \mathring{u}$  means round upwards.

end for

#### 4.2.6.1.2. Parameters for turbo codes

'X': systematic bits sequence of the upper constituent encoder s.t.  $(x_1, x_2, \dots, x_k, \dots)$  in section 4.2.3.2.1.

'Y': parity bits sequence of the upper constituent encoder s.t.  $(y_1, y_2, \dots, y_k, \dots)$  in section 4.2.3.2.1.

'Z': parity bits sequence of the lower constituent encoder s.t.  $(z_1, z_2, \dots, z_k, \dots)$  in section 4.2.3.2.1.

If repetition is to be performed, such as  $DN_{ij} \geq 0$ , parameters for turbo codes are the same as parameter for convolutional codes.

If puncturing is to be performed, parameters are as follows.

$$N_{cs} = \mathring{e}N_{ij}/3\mathring{u} \text{ where } \mathring{e} \mathring{u} \text{ means round downwards.}$$

$N_{is} = N_{cs}$  for 'X' sequence in rate matching algorithm block 1.

$N_{is} = N_{cs} - \mathring{e}DN_{ij}/2\mathring{u}$  for 'Y' sequence in rate matching algorithm block 2 where  $\mathring{e} \mathring{u}$  means round downwards.

$N_{is} = N_{cs} - \mathring{e}'DN_{ij}/2\mathring{u}$  for 'Z' sequence in rate matching algorithm block 3 where  $\mathring{e}' \mathring{u}$  means round upwards.

$q = \mathring{e}N_{cs} / (\mathring{O}N_{is} - N_{cs})\mathring{O}\mathring{u}$ ;  $q$  is the average puncturing distance of 'Y' or 'Z' sequence

(a,b)=(2,1) for the rate matching algorithm block 2.  
(a,b)=(1,1) for the rate matching algorithm block 3.

```

if(q ≤ 2)
  for l=0 to Fi-1
    if('Y' sequence)
      S[lF[(3l+1) mod Fi]] = 1 mod 2;
    if('Z' sequence)
      S[lF[(3l+2) mod Fi]] = 1 mod 2;
    end for
  else
    if q is even
      then q' = q - gcd(q, Fi) / Fi -- where gcd(q, Fi) means greatest common divisor of q and Fi
      -- note that q' is not an integer, but a multiple of 1/8
    else q' = q
    endif
    for l=0 to Fi - 1
      r = εl * q' ÷ mod Fi;
      if('Y' sequence)
        S[lF[(3r+1) mod Fi]] = εl * q' ÷ div Fi;
      if('Z' sequence)
        S[lF[(3r+2) mod Fi]] = εl * q' ÷ div Fi;
      endfor
    endif
  
```

#### 4.2.6.2. Determination of rate matching parameters in downlink

For downlink  $N_{data,j}$  does not depend on the transport format combination  $j$ .  $N_{data,j}$  is given by the channelization code(s) assigned by higher layers.

NOTE: The rule to convert the rate matching attributes in downlink to the parameters input to rate matching pattern algorithm are working assumption. So, it remains to be verified that they hold for all possible transport format combinations. It has been identified that the case when the transport format combination with highest rate include a transport format with zero bits need special treatment.

Radio frame segmentation is performed after 1<sup>st</sup> interleaving and  $N_{ij}$  is therefore calculated as:

$$l = TF_i(j) \quad \text{and} \quad N_{ij} = \left\lceil \frac{N_{i,l}^{TI}}{F_i} \right\rceil$$

The number of bits repeated or punctured,  $DN_{iL}$ , within one radio frame for each transport channel is calculated for the transport format combination  $L$  with highest bitrate with the relations given in Section 4.2.6.

If fix positions of the transport channels in the radio frame are used then the same  $DN_{ij}$  is used for all transport format combinations and the last part of the rate-matching pattern omitted. That is to say for all transport format combinations  $j$  we have:

$$\Delta N_{ij} = \Delta N_{iL}$$

When flexible positions of the transport channels are used, the number of bits  $DN_{ij}$  repeated or punctured for all transport format combinations  $j$  other than  $L$  is calculated as:

$$\Delta N_{ij} = \left\lceil \frac{\Delta N_{iL}}{N_{iL}} \cdot N_{ij} \right\rceil$$

For each transmission time interval, the rate-matching pattern is calculated with the algorithm in Section 4.2.6.3. The following parameters are used as input:

$$l = TF_i(j) \quad \text{and} \quad \Delta N = \Delta N_{il}^{TTI} = F_i \Delta N_{ij}$$

$$N = N_{il}^{TTI}$$

S=0.

#### 4.2.6.2.1. Parameter for convolutional codes

'X' coded bits sequence from output 0 of convolutional coder in section 4.2.3.1.1.

'Y' coded bits sequence from output 1 of convolutional coder in section 4.2.3.1.1.

'Z' coded bits sequence from output 2 of convolutional coder in section 4.2.3.1.1.

$$N_{cs} = N$$

$$N_{is} = N + DN$$

$(a,b)=(2,1)$  for the rate matching algorithm block 1.

#### 4.2.6.2.2. Parameter for turbo codes

'X': systematic bits sequence of the upper constituent encoder s.t.  $(x_1, x_2, \dots, x_k, \dots)$  in section 4.2.3.2.1.

'Y' : parity bits sequence of the upper constituent encoder s.t.  $(y_1, y_2, \dots, y_k, \dots)$  in section 4.2.3.2.1.

'Z' : parity bits sequence of the lower constituent encoder s.t.  $(z_1, z_2, \dots, z_k, \dots)$  in section 4.2.3.2.1.

If repetition is to be performed, such as  $DN > 0$ , parameters for turbo codes are the same as parameter for convolutional codes.

If puncturing is to be performed, parameters are as follows.

$N_{cs} = \bar{e}N/3\hat{u}$  where  $\bar{e}$  means round downwards.

$N_{is} = N_{cs}$  for 'X' sequence in rate matching algorithm block 1.

$N_{is} = N_{cs} - \bar{e}DN/2\hat{u}$  for 'Y' sequence in rate matching algorithm block 2 where  $\bar{e}$  means round downwards.

$N_{is} = N_{cs} - \hat{e}DN/2\hat{u}$  for 'Z' sequence in rate matching algorithm block 3 where  $\hat{e}$  means round upwards.

$(a,b)=(2,1)$  for the rate matching algorithm block 2.

$(a,b)=(1,1)$  for the rate matching algorithm block 3.

Systematic bit should not be punctured .

### 4.2.6.3. Rate matching algorithm

All rate matching algorithm blocks in Figure 4-1 and 4-2 in section 4.2.6 use the same rate matching algorithm below with different parameters for convolutional codes and turbo codes, respectively.

For convolutional codes, in Figure 4-1 and 4-2, DEMUX is always at the position 1 and parameters for rate matching algorithm block 1 are fixed to the value given in section 4.2.6.2.1 or 4.2.6.3.1.

For turbo codes, in Figure 4-1 and 4-2, DEMUX repeats switching for puncturing or is always at the position 1 for repetition, respectively. Parameters for rate matching algorithm blocks are fixed to the value given in section 4.2.6.2.2 or 4.2.6.3.2.

Denote the bits before rate matching by:

$$c_1, c_2, c_3, \dots, c_{N_{cs}}$$

The common rate matching rule for both convolutional codes and turbo codes is as follows:

if puncturing is to be performed

$$y = \bar{D}N_{cs} - N_{is}$$

```

e = (2a*S(k) * y + b*Ncs) mod 2a*Ncs -- initial error between current and desired puncturing ratio
if e=0 then e=a*Ncs
m = 1 -- index of current bit
do while m <= Ncs
    e = e - 2a*y -- update error
    if e <= 0 then -- check if bit number m should be punctured
        puncture bit cm
        e = e + 2a*Ncs -- update error
    end if
    m = m + 1 -- next bit
end do
else
y = DN / (Ncs - Nis)

e = (2a*S(k) * y + b*Ncs) mod 2a*Ncs -- initial error between current and desired puncturing ratio
if e=0 then e=a*Ncs
m = 1 -- index of current bit
do while m <= Ncs
    e = e - 2a*y -- update error
    do while e <= 0 -- check if bit number m should be repeated
        repeat bit cm
        e = e + 2a*Ncs -- update error
    enddo
    m = m + 1 -- next bit
end do
end if

```

A repeated bit is placed directly after the original one.

---

### 3. Text proposal for TS 25.222

----- Start text proposal -----

#### 6.2.3. Channel coding

The following channel coding schemes can be applied to transport channels.

- Convolutional coding
- Turbo coding
- No channel coding

The maximum encoding segment length for turbo coding is 5120 bits. In Real Time (RT) services a FEC coding is used, instead Non Real Time (NRT) services could be well managed with a proper combination of FEC and ARQ.



**Table 6.2.3-1 Error Correction Coding Parameters**

Transport channel type	Coding scheme	Coding rate
BCH	Convolutional code	1/2
PCH		
FACH		
RACH		
DCH	Turbo code	1/2, [2/3, 7/8] <Editor's note: the values in square brackets have not yet been approved.>
DCH		

Note 1: The exact physical layer encoding/decoding capabilities for different code types are FFS.

Note 2: In the UE the channel coding capability should be linked to the terminal class.

----- End text proposal -----

----- Start text proposal -----

### 6.2.3.2.1. Turbo coder

<Note: 4-state SCCC is not included in Release -99.>

<Note: It needs to be clarified from TSG SA what are the service specifications with respect to different qualities of service. The performance below BER of  $10^{-6}$  needs to be studied if there is a requirement for this quality of services over the physical layer.>

For data services requiring quality of service between  $10^{-3}$  and  $10^{-6}$  BER inclusive, parallel concatenated convolutional code (PCCC) with 8-state constituent encoders is used.

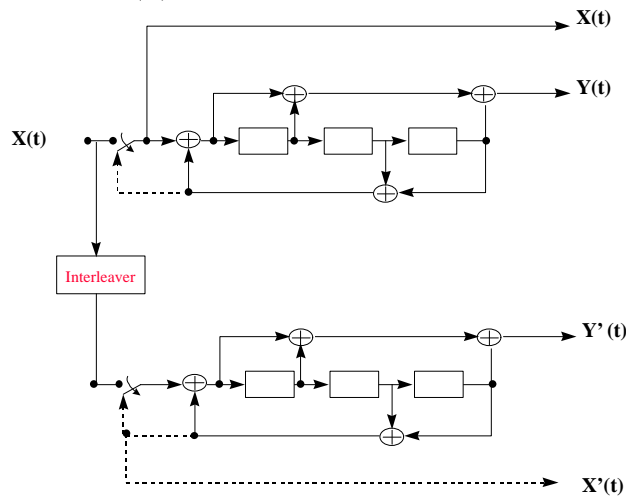
The transfer function of the 8-state constituent code for PCCC is

$$G(D) = \left[ 1, \frac{n(D)}{d(D)} \right]$$

where,

$$d(D) = 1 + D^2 + D^3$$

$$n(D) = 1 + D + D^3$$



**Figure 6-3. Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)**

The initial value of the shift registers of the PCCC encoder shall be all zeros.

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate 1/3 ~~or 1/2~~. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc. ~~For rate 1/2, the parity bits produced by the constituent encoders are alternately punctured to produce the output sequence X(0), Y(0), X(1), Y'(1), X(2), Y(2), X(3), Y'(3), etc.~~

The SCCC is a rate 1/3 SCCC, The outer code of the SCCC is a rate 2/3 obtained by puncturing a rate 1/2 code with generating matrix

$$G^{(o)}(Z) = (1, (1 + Z^2)/(1 + Z + Z^2))$$

The rate 2/3 is obtained by puncturing every other parity-check bit. The inner code is a rate 1/2 systematic recursive convolutional code with the same previous generating matrix

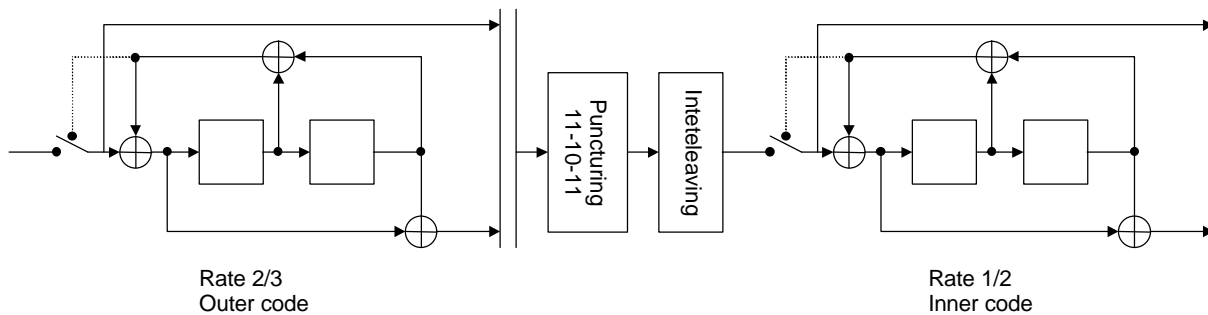


Figure 6-4. Structure of the 4 state SCCC encoder (dotted line effective for trellis termination only).

----- End proposal -----

----- Start text proposal -----

### 6.2.6. Rate matching

Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to repeat or puncture is calculated.

The number of bits on a transport channel can vary between different transmission time intervals. When the number of bits between different transmission time intervals in uplink is changed, bits are repeated to ensure that the total bit rate after second multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

$N_{ij}$ : Number of bits in a radio frame before rate matching on transport channel  $i$  with transport format combination  $j$ .

$\Delta N_{ij}$ : If positive - number of bits that should be repeated in each radio frame on transport channel  $i$  with transport format combination  $j$ .

If negative - number of bits that should be punctured in each radio frame on transport channel  $i$  with transport format combination  $j$ .

$RM_i$ : Semi-static rate matching attribute for transport channel  $i$ . Signalled from higher layers.

$PL$ : Puncturing limit for uplink. This value limit the amount of puncturing that can be applied in order to avoid multicode or enable the use of a higher spreading factor. Signalled from higher layers.

$N_{data,j}$ : Total number of bits that are available for a CCTrCH in a radio frame with transport format combination  $j$ .

$T$ : Number of transport channels in a CCTrCH.

$Z_{mj}$ : Intermediate calculation variable.

$F_i$ : Number of radio frames in the transmission time interval of transport channel  $i$ .

$k$ : Radio frame number in the transmission time interval of transport channel  $i$  ( $0 \leq k < F_i$ ).

$q$ : Average puncturing distance.

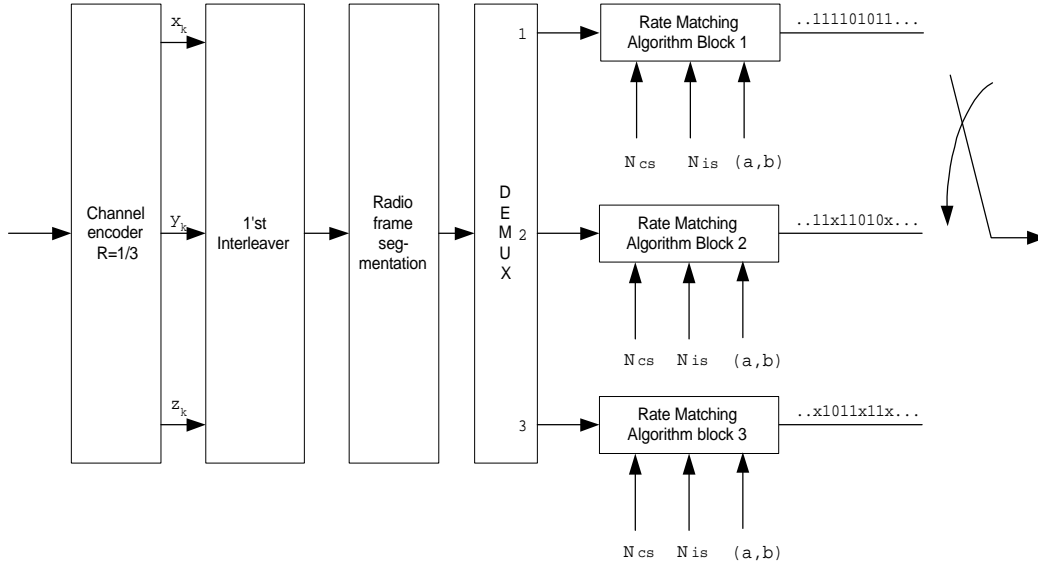
$I_F(k)$ : The inverse interleaving function of the 1<sup>st</sup> interleaver (note that the inverse interleaving function is identical to the interleaving function itself for the 1<sup>st</sup> interleaver).

$S(k)$ : The shift of the puncturing pattern for radio frame k.

$TF_i(j)$ : Transport format of transport channel  $i$  for the transport format combination  $j$ .

$N_{cs}$ : Number of bits before rate matching for each rate matching algorithm block.

$N_{is}$ : Number of bits after rate matching for each rate matching algorithm block.



**Figure 6-1. Overall rate matching block diagram after first interleaving where  $x$  denotes punctured bit.**

### 6.2.6.1. Determination of rate matching parameters

The following relations are used when calculating the rate matching pattern:

$$Z_0 = 0$$

$$Z_{mj} = \left\lfloor \frac{\sum_{i=1}^m RM_i \cdot N_{ij}}{\sum_{i=1}^T RM_i \cdot N_{ij}} \cdot N_{data} \right\rfloor \quad \text{for all } m = 1 \dots T, \text{ where } \lfloor \cdot \rfloor \text{ means round downwards}$$

$$\Delta N_{ij} = Z_{ij} - Z_{i-1,j} - N_{ij} \quad \text{for all } i = 1 \dots T$$

Puncturing can be used to minimise the number of required transmission capacity. The maximum amount of puncturing that can be applied is signalled at connection setup from higher layers and denoted by  $PL$ . The possible values for  $N_{data}$  are always multiples of the dedicated physical channel with the smallest capacity, reduced by the amount of bits which carry the TFCI. The supported set of  $N_{data}$ , denoted  $SET0$ , depends on the UE capabilities.  $N_{data,j}$  for the transport format combination  $j$  is determined by executing the following algorithm:

$$SET1 = \left\{ N_{data} \text{ in } SET0 \text{ such that } N_{data} - PL \sum_{i=1}^T \frac{RM_i}{\min\{RM_l\}} N_{ij} \text{ is non-negative} \right\}$$

$$N_{data,j} = \min SET1$$

The number of bits to be repeated or punctured,  $\Delta N_{ij}$ , within one radio frame for each transport channel  $i$  is calculated with the relations given at the beginning of this section for all possible transport format combinations  $j$  and selected every radio frame.

For each radio frame, the rate matching pattern is calculated with the algorithm in Section 6.2.5.2, where  $DN = DN_{ij}$  and  $N = N_{ij}$ .

Additionally, the following parameters are needed:

#### 6.2.6.1.1. Parameters for convolutional codes

'X' coded bits sequence from output 0 of convolutional coder in section 6.2.3.1.1.

'Y' coded bits sequence from output 1 of convolutional coder in section 6.2.3.1.1.

'Z' coded bits sequence from output 2 of convolutional coder in section 6.2.3.1.1.

$$N_{cs} = N_{ij}$$

$$N_{is} = N_{ij} + DN_{ij}$$

$(a,b)=(2,1)$  for the rate matching algorithm block 1.

$$q = \lfloor N_{ij} / (|\Delta N_{ij}|) \rfloor \text{ where } \lfloor \rfloor \text{ means round downwards, and } | \cdot | \text{ means absolute value.}$$

If  $q$  is even

Then  $q' = q - \text{gcd}(q, F_i)/F_i$  -- where  $\text{gcd}(q, F_i)$  means greatest common divisor of  $q$  and  $F_i$  -- note that  $q'$  is not an integer, but a multiple of 1/8.

Else

$$q' = q$$

endif

for  $l=0$  to  $F_i - 1$

$$S(I_F([l * q'] \bmod F_i)) = ([l * q'] \text{div} F_i) \text{ -- where } \lceil \rceil \text{ means rounding upwards.}$$

end for

#### 6.2.6.1.2. Parameters for turbo codes

'X': systematic bits sequence of the upper constituent encoder s.t.  $(x_1, x_2, \dots, x_k, \dots)$  in section 6.2.3.2.1.

'Y': parity bits sequence of the upper constituent encoder s.t.  $(y_1, y_2, \dots, y_k, \dots)$  in section 6.2.3.2.1.

'Z': parity bits sequence of the lower constituent encoder s.t.  $(z_1, z_2, \dots, z_k, \dots)$  in section 6.2.3.2.1.

If repetition is to be performed, such as  $DN_{ij} \geq 0$ , parameters for turbo codes are the same as parameter for convolutional codes.

If puncturing is to be performed, parameters are as follows.

$$N_{cs} = \hat{e} N_{ij} / 3 \hat{u} \text{ where } \hat{e} \hat{u} \text{ means round downwards.}$$

$$N_{is} = N_{cs} \text{ for 'X' sequence in rate matching algorithm block 1.}$$

$$N_{is} = N_{cs} - \hat{e} DN_{ij} / 2 \hat{u} \text{ for 'Y' sequence in rate matching algorithm block 2 where } \hat{e} \hat{u} \text{ means round downwards.}$$

$$N_{is} = N_{cs} - \hat{e} DN_{ij} / 2 \hat{u} \text{ for 'Z' sequence in rate matching algorithm block 3 where } \hat{e} \hat{u} \text{ means round upwards.}$$

$$q = \hat{e} N_{cs} / (\hat{e} N_{is} - N_{cs} \hat{e}) \hat{u}; \text{ } q \text{ is the average puncturing distance of 'Y' or 'Z' sequence}$$

$(a,b)=(2,1)$  for the rate matching algorithm block 2.

$(a,b)=(1,1)$  for the rate matching algorithm block 3.

if  $(q \leq 2)$

for  $l=0$  to  $F_i - 1$

if ('Y' sequence)

$$S[I_F[(3l+1) \bmod F_i]] = 1 \bmod 2;$$

if ('Z' sequence)

$$S[I_F[(3l+2) \bmod F_i]] = 1 \bmod 2;$$

end for

else

if  $q$  is even

then  $q' = q - \text{gcd}(q, F_i)/F_i$  -- where  $\text{gcd}(q, F_i)$  means greatest common divisor of  $q$  and  $F_i$

-- note that  $q'$  is not an integer, but a multiple of 1/8

else  $q' = q$

```

endif
for l=0 to Fi-1
  r =  $\hat{e}l * q' \hat{u} \bmod F_i$ ;
  if ('Y' sequence)
     $S[l_F[(3r+1) \bmod F_i]] = \hat{e}l * q' \hat{u} \bmod F_i$ ;
  if ('Z' sequence)
     $S[l_F[(3r+2) \bmod F_i]] = \hat{e}l * q' \hat{u} \bmod F_i$ ;
  endif
endif
endif

```

## 6.2.6.2. Rate matching algorithm

All rate matching algorithm blocks in Figure 6-1 in section 6.2.6 use the same rate matching algorithm below with different parameters for convolutional codes and turbo codes, respectively.

For convolutional codes, in Figure 6-1, DEMUX is always at the position 1 and parameters for rate matching algorithm block 1 are fixed to the value given in section 6.2.6.2.1 or 6.2.6.3.1.

For turbo codes, in Figure 6-1, DEMUX repeats switching for puncturing or is always at the position 1 for repetition, respectively. Parameters for rate matching algorithm blocks are fixed to the value given in section 6.2.6.2.2 or 6.2.6.3.2.

Denote the bits before rate matching by:

$c_1, c_2, c_3, \dots, c_{N_{CS}}$

The common rate matching rule for both convolutional codes and turbo codes is as follows:

if puncturing is to be performed

```

y =  $\frac{-AN_{CS} - N_{IS}}{2a}$ 
e = ( $2a * S(k) * y + b * N_{CS}$ ) mod  $2a * N_{CS}$   -- initial error between current and desired puncturing ratio
if e=0 then e=a*NCS
m = 1  -- index of current bit
do while m <= NCS
  e = e -  $2a * y$   -- update error
  if e <= 0 then  -- check if bit number m should be punctured
    puncture bit cm
    e = e +  $2a * N_{CS}$   -- update error
  end if
  m = m + 1  -- next bit
end do
else
y =  $\frac{AN - (N_{CS} - N_{IS})}{2a}$ 
e = ( $2a * S(k) * y + N_{CS}$ ) mod  $2N_{CS}$   -- initial error between current and desired puncturing ratio
if e=0 then e=a*NCS
m = 1  -- index of current bit
do while m <= NCS
  e = e -  $2a * y$   -- update error
  do while e <= 0  -- check if bit number m should be repeated
    repeat bit cm
    e = e +  $2a * N_{CS}$   -- update error
  enddo
  m = m + 1  -- next bit
end do
end if

```

A repeated bit is placed directly after the original one.

---

## 4. References

- [1] TS 25.212 (V2.0.1): “Multiplexing and channel coding (FDD)”
- [2] TS 25.222 (V2.0.2): “Multiplexing and channel coding (TDD)”
- [3] “Text proposal for rate matching algorithm for turbo codes in S25.212, S25.222”, Samsung Electronics Co. and LGIC, WG1#7, R1-99b56.